PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#18
LDJ
2-20-04

Applicant(s): Marc Tremblay and William Joy

Title: EFFICIENT HANDLING OF A LARGE REGISTER FILE FOR CONTEXT SWITCHING AND FUNCTION CALLS AND RETURNS

| | | | | |
|---|---|---|---|---|
| Application No.: | 09/812,733 | Filed: | March 19, 2001 | |
| Examiner: | David Y. Eng | Group Art Unit: | 2155 | |
| Atty. Docket No.: | 004-2612-1 | | | |



January 20, 2004

Mail Stop Appeal Brief - Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313

# APPELLANT'S BRIEF (37 C.F.R. § 1.192)

This brief is in furtherance of the Notice of Appeal, filed on June 19, 2003. The fees required under § 1.17(c), are provided in the accompanying Transmittal. A Petition for Extension of Time accompanies this Brief, extending the period for filing until January 20, 2004. This brief is being transmitted in triplicate pursuant to 37 C.F.R. § 1.192(a).

## REAL PARTY IN INTEREST

The real party in interest in this appeal is Sun Microsystems, Inc., as evidenced by the assignment recorded at Reel 9619/ Frame 0646 (recorded in the parent application to which the instant application claims priority as a continuation).



## RELATED APPEALS AND INTERFERENCES

Appellants have no knowledge of any related appeals or interferences.

## STATUS OF CLAIMS

Claims 29-38 are presented herein on appeal. Of those, applicant seeks to cancel claims 30-32 and 36 by amendment (unentered and transmitted herewith). (All claim numbers indicated

herein are consistent with the Examiner's January 15, 2003 renumbering of claims.) All claims (29-38) were rejected in a final Office action dated January 15, 2003. That final rejection is now appealed. All claims presented on appeal are reproduced in the Appendix attached hereto.

## STATUS OF AMENDMENTS

A response after final was filed and an advisory action followed. In addition to dismissing Applicants argument that the extant statutory (35 U.S.C. § 101) double patenting rejection was unsustainable, the advisory (mailed April 1, 2003) refused entry of amendments that sought to:

(1) cancel claims 31 and 32; and
(2) eliminate an element from claim 33, thereby avoiding correspondence with elements of an issued claim of and eliminating a double patenting issue for appeal.

Applicant presents these amendments again (by way of a second and accompanying amendment under 37 C.F.R. § 1.116). In addition, a terminal disclaimer is filed herewith, thereby overcoming the non-statutory (obviousness-type) double patenting rejection of claim 33.

## SUMMARY OF INVENTION

The present invention relates to context switching in a processor. More specifically, the present invention will be understood in the context of efficient handling of a large register file for context switching. A large register file has many advantages but also has several drawbacks. For example, a processor with a large register file (or other similar storage) can incur large overhead during context switching since values held for a given process or thread can reside in a large number of registers. Context switching may include relocating values (for a first process or thread) stored in such registers to a context store and relocating values (for a second process or thread) from such a context store to the register file. Since context switching overhead can adversely affect processor performance, efficient techniques are desired.

Context switching can be facilitated in a processor that includes a large register file in part by utilizing dirty bit storage associated with the register file and dirty bit logic that controls resetting of the dirty bit storage. The dirty bit logic determines whether a register or group of

registers in the register file has been written since the process was loaded or the context was last restored and, if written, generates a value in the dirty bit storage that designates the written condition of the register or group of registers. When the context is next saved, the dirty bit logic saves a particular register or group of registers if the dirty bit storage indicates that a register or group of registers was written. If the register or group of registers was not written, the context is switched without saving the register or group of registers. The dirty bit storage can be initialized when a process is loaded or the context changes.

At least two variations on a context switch controller configuration are illustrated, respectively, in FIGS. 5 and 6. Corresponding description appears throughout the Specification, but particularly beginning at page 16, line 19 and continuing through page 19, line 4.

Referring to FIG. 5, one embodiment of a context switch controller and/or related techniques exploits a register file 510 and dirty bit structures including a dirty bit storage 512 and a dirty bit logic 514. In an illustrative example, an instruction register 515 stores a destination register (rd) field 516 that is used to address registers in register file 510. The dirty bit logic 514 accesses a portion of the rd field 516 (e.g., the three most significant bits thereof), thereby classifying registers of the register file 510 into respective groups, e.g., octants, of register segments. For example, register file 510 segmented using the upper three bits of the rd field 516 is classified into eight octants including octants 1 through 7 containing addresses 0-31, 32-63, 64-95, 96-127, 128-159, 160-191, 192-223, and 224-255, respectively.

The large register file 510, which can operate as an executive storage, uses the dirty bit storage 512 and the dirty bit logic 514 to control context switching of the processor (see processor 100, FIG. 1). In an illustrative embodiment, the dirty bit storage 512 includes a dirty bit register that holds an access history of registers in the register file 510. The dirty bit logic 514 controls setting and resetting of bits in the dirty bit storage 512.

The dirty bit logic 514 monitors various data load and data store instructions executed by a general functional unit (e.g., GFU 222, FIG. 2) to determine whether a register or group of registers in the register file 510 has been written since an executing process was loaded or since the context was last restored. If the register or register group has been written, the dirty bit logic 514 generates a value in the dirty bit storage 512 that designates the written condition of the

register or group of registers. The next time the processor context is saved, the dirty bit logic 514 saves a particular register or group of registers when the dirty bit storage 512 indicates that a register or group of registers was written. If the register or group of registers was not written, the context is switched without saving the register or group of registers. The dirty bit storage 512 is initialized when an executable process is loaded or the context changes.

Referring to FIG. 6, another embodiment of a context switching controller and/or related techniques exploits dirty bit enable storage 620 in combination with the register file 610 and dirty bit structures including dirty bit storage 612 and dirty bit logic 614 that controls resetting of the dirty bit storage 612. The dirty bit logic 614 determines whether a register or group of registers in the register file 610 has been written since the context was last restored. If the register or group was written, the dirty bit logic 614 generates a value in the dirty bit storage 612 that designates the written condition of the register or group of registers. When the context is next saved, the dirty bit logic 614 saves a particular register or group of registers when the dirty bit storage 612 indicates that a register or group of registers was written. The dirty bit enable storage 620 is used to enable or disable the dirty bit operation on a group-by-group basis within the register file 610. The dirty bit enable storage 620 has a bit that corresponds to each of the bits in the dirty bit storage 612. The dirty bit enable storage 620 has bit values that are programmed to disable or enable access to the register or group of registers that correspond to each bit.

ISSUES

1. The outstanding statutory (35 U.S.C. § 101) double patenting rejection under 35 U.S.C. § 101 of claims 29, 34, 35, 37 and 38 must be reversed.

2. The outstanding non-statutory (obviousness-type) double patenting rejection of claim 33 (and any other claims) is overcome a terminal disclaimer filed herewith.

3. The Office has not made out a *prima facie* case of obviousness with respect to claims 34, 37 or 38.

GROUPING OF CLAIMS

| | |
|---|---|
| Group I: | Claim 29 |
| Group II: | Claim 33 |
| Group III: | Claims 34, 37 and 38 |
| Group IV: | Claim 35 |

ARGUMENTS

*Statutory Double Patenting -Rejections under 35 USC §101*

Claims of Group I (claim 29), Group III (claims 34, 37 and 38) and Group IV (claim 35) have been rejected under 35 USC §101 as claiming the same invention as that of claims 1 and 11-12 of prior U.S. Patent No. 6,205,543. Applicants respectfully traverse these rejections.

*Group I*: In rejecting claim 29, the Office has alleged correspondence between a single claim term of claim 29 ("operand data storage") and another claim term of claim 1 ("executive storage") of claim 1 of issued U.S. Patent 6,205,543. While applicant certainly does not acquiesce in the Office's interpretation, correspondence (if any) of such terms is irrelevant. Subject matter of claim 29 differs from that of claim 1 of U.S. Patent No. 6,205,543 in at least the following substantial way. Claim 29 of the application does not include the limitation requiring that "dirty bit logic" is "`responsive to a context switch by saving storage groups based on the evaluation of the classified destinations`" as recited in claim 1 of the patent.

The Examiner alleges that Applicants have "fail[ed] to provide any arguments as to why the elements of the rejected claims do not correspond to the elements of the patented claims." Such allegation is completely without merit. The Applicants respectfully note that in responding to both the preceding non-office action and the outstanding final action, not only did Applicants identify a specific limitation of patented claim 1 not recited in presently rejected claim 29, but also identify a specific legal error, namely failure to provide the literal infringement analysis required to establish a prima facie case for same-invention double patenting (*See* In re Vogel 164 U.S.P.Q. 619 and MPEP § 804). Examiner's statutory double patenting rejection was (and is,

now again) properly traversed. Applicant respectfully notes that the proper legal standard for statutory double patenting is identity not mere correspondence.

Although the Office's reliance on claims 11-12 of issued U.S. Patent 6,205,543 is unclear (though presumably targeted at claims of Group III), Applicant notes that claims 11-12 of issued U.S. Patent 6,205,543 also recite limitations, be they "`a plurality of functional units`" or details of dirty bit logic operation (element 4 of claim 11), simply not included in claim 1.

Applicants therefore request that this Honorable board reverse the rejection of claim(s) of Group I and direct that a patent reciting such claims be issued forthwith.

*Group III*: In rejecting claims 34, 37 and 38, the Office has simply failed to make out a *prima facie* case. No correspondence of elements (identical elements or otherwise) appears in the outstanding final rejection. For this reason alone the statutory double patenting rejection should be reversed. However, Applicant further notes that (as with claim 29), claim 34 (and those dependent therefrom) do not include a limitation requiring that "dirty bit logic" is "`responsive to a context switch by saving storage groups based on the evaluation of the classified destinations`" as recited in claim 1 of the patent. The limitation is also absent from claims 35, 37 and 38. Claims 11-12 of issued U.S. Patent 6,205,543 also recite limitations, be they "`a plurality of functional units`" or details of dirty bit logic operation (element 4 of claim 11), simply not included in claim 34, 37 or 38.

Applicants therefore request that this Honorable board reverse the rejection of claim(s) of Group III, and direct that a patent reciting such claims be issued forthwith.

*Group IV*: Claim 35 is rejected only on grounds of same invention double-patenting (35 U.S.C. § 101). In rejecting claim 35, the Office has simply failed to make out a *prima facie* case. No correspondence of elements (identical elements or otherwise) appears in the outstanding final rejection. For this reason alone the statutory double patenting rejection should be reversed. However, Applicant further notes that claim 34 (from which claim 35 depends) does not include a limitation requiring that "dirty bit logic" is "`responsive to a context switch by saving storage groups based on the evaluation of the classified`

destinations" as recited in claim 1 of the patent. The limitation is also absent from claim 35. Claims 11-12 of issued U.S. Patent 6,205,543 also recite limitations, be they "a plurality of functional units" or details of dirty bit logic operation (element 4 of claim 11), simply not included in claim 35. Claim 35 of the present application and claim 1 of U.S. Patent 6,205,543 do not claim identical subject matter.

Applicants therefore request that this Honorable board reverse the rejection of claim(s) of Group IV, and direct that a patent reciting such claims be issued forthwith.

*Rejections under Non-statutory Obviousness-type Double Patenting*

*Group II*: Claim 33 (and only claim 33) has been rejected under the judicially created doctrine of double patenting over claims 1 and 20 of U.S. Patent No. 6,408,383. Without acquiescing in the propriety of the rejection, Applicants have submitted a terminal disclaimer in compliance with 37 CFR § 1.321(c). Accordingly the rejection of claim 33 has been overcome and Applicants request that this Honorable board direct that a patent reciting claim(s) of Group II be issued forthwith.

*Rejections under 35 U.S.C. § 103(a)*

*Group III*: Claim 34 and 37-38 have been rejected as obvious over U.S. Patent No. 6,470,433 to Emer et al. (*Emer*). While the term "context switch controller" is positively recited in the claim, the Office has effectively discounted the term. In particular, the Examiner assumes that *Emer's* disclosure of a processor including registers with associated valid bits teaches or suggests a context switch controller that employs dirty bits in the way claimed by Applicants. Careful reading of *Emer* makes clear that no such use for context switch control is disclosed. Indeed, Examiner relies on general background description of thread context switching (col. 1, line 36) in a CPU and improperly mates that description with totally unrelated description of register associated valid bits (col. 5, lines 3-6). While *Emer* details (beginning at column 6, line 18) a variety of schemes for selection of a thread (or threads) for which instructions are to be fetched (e.g., based on modified round-robin techniques, branch mispredict or cache miss predictions, or counts of instructions in particular pipeline stages), nothing in *Emer* discloses or suggests, taken alone or in combination with other art of record, a context switch

controller that employs dirty (or valid) bits. There is simply no relation, in *Emer*, between thread context switch control and valid bits. Valid bits are employed in *Emer* to signal that an operand of an instruction is available and that the instruction can be issued. Valid bits have nothing at all to do with context switch control. Accordingly, no switch controller is disclosed or suggested that employs dirty bit storage and dirty bit logic as claimed. No prima facie case of obviousness has been made.

While claim 34 (as pending, and unamended) recites a "context switch controller" (as claimed) that includes particular recited elements, namely dirty bit storage and dirty bit logic (employed as claimed), applicants have presented an amendment (herewith) to emphasize relation between "the context switch controller" and "dirty bit storage" and to avoid possible misinterpretation of claim as reciting elements of a processor, without relation to the "context switch controller." The amendment requires only cursory review, presents no new issues, and entry is respectfully requested.

With or without the amendment, claim 34 and those dependent therefrom are all allowable over *Emer* for the reasons outlined above and a notice to that effect is respectfully requested. Accordingly the rejection of claims 34, 37 and 38 has been overcome and Applicants request that this Honorable board direct that a patent reciting claim(s) of Group III be issued forthwith.

## CONCLUSION

For the at least the foregoing reasons, Appellants' presently claimed invention does not represent double patenting and would not have been obvious to one of ordinary skill in the art under 35 U.S.C. § 103(a) in view of the cited prior art. Accordingly, this Honorable Board is respectfully requested to reverse the rejections of Claims 29, 33, 34, 35, 37 and 38 and to direct the claims of the present application to be issued.

**CERTIFICATE OF MAILING OR TRANSMISSION**

I hereby certify that, on the date shown below, this correspondence is being

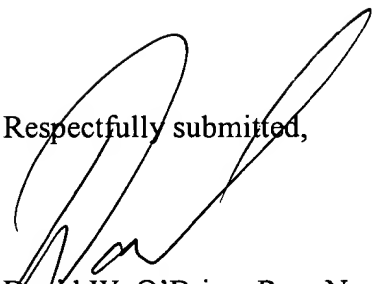
☐ deposited with the US Postal Service with sufficient postage as first class mail, in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

☐ facsimile transmitted to the US Patent and Trademark Office.

David W. O'Brien _____ Date _____

**EXPRESS MAIL LABEL:** EV401038895US

Respectfully submitted,



David W. O'Brien, Reg. No. 40,107
Attorney for Applicant(s)
(512) 338-6314
(512) 338-6301 (fax)

## APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

29. A context switch controller in a processor that includes an operand data storage for holding data operated upon by instructions executing on the processor, the operand data storage being divided into a plurality of storage groups containing one or more storage elements, the context switch controller comprising:

a dirty bit storage including one or more storage bits that correspond to one or more respective storage groups in the operand data storage; and

a dirty bit logic coupled to the dirty bit storage and coupled to receive a destination address field of the instructions, the dirty bit logic responsive to an executed instruction by classifying a destination access as a targeted storage group according to information in the destination address field of the executed instruction and by evaluating the classified destination based on whether the instruction updates the targeted storage group.

33. A context switching logic in a processor that includes an executive storage for holding operand data operated upon by instructions executing on the processor, the executive storage being divided into a plurality of storage groups containing one or more storage elements, the context switching logic comprising:

means for utilizing a dirty bit storage including a plurality of storage bits corresponding to a plurality of respective storage groups in the executive storage;

means for receiving a destination address field of the executing instructions;

means for responsive to an executed instruction, classifying a destination access as a targeted storage group according to information in the destination address field of the executed instruction;

means for evaluating the classified destination based on whether the instruction updates the targeted storage group; and

means responsive to a context switch for saving storage groups based on the evaluation of the classified destinations.

34. A context switch controller in a processor comprising:

a data storage unit divided into a plurality of storage groups;

a dirty bit storage coupled to the data storage and including one or more storage bits corresponding to one or more respective storage groups in the data storage unit; and

a dirty bit logic coupled to the dirty bit storage and configured to receive a destination address of one or more instructions executing on the processor.

35. The context switch controller of claim 34, wherein the dirty bit logic is responsive to one or more instructions executed on the processor to

classify a destination access as a targeted storage group according to information in a destination address field of the one or more instructions executed on the processor; and

evaluate the classified destination address based on whether the instruction updates the targeted storage group.

37. The context switch controller of claim 34, wherein the data storage unit is configured to hold data operated by the one or more instructions executed on the processor.

38. The context switch controller of claim 34, wherein each one of the plurality of storage groups comprises one or more storage elements.